# Evaluating and optimising the cost of coupling in the UKC2 regional system

Forecasting Research
Technical Report No: 621

July 2017

Juan Manuel Castillo Sanchez
Huw Lewis

If printing double sided you will need this blank page.  If printing single sided, please delete this page.

# 1. Introduction

This Technical Report documents worked aimed at exploring the different set of parameters that can be modified in the UK Environmental Prediction Prototype suites to improve their computational performance, without altering their science results. The aim of performance optimization is to produce science results as fast as possible (in terms of real-world run time, or wall time) with the minimum supercomputing resources. Estimates are also required to provide measures of the cost associated with coupling, in order to provide current benchmarks and prioritise future development. Lewis et al. (2017) further detail on the technical configurations and suite structures used in this study.

Considering that there are three completely different modelling system components (atmosphere, ocean, wave) running in both coupled and uncoupled mode within the UKC2 research systems, the number of modifiable parameters and their possible values is quite large, in particular: code version, compiler version, compilation parameters, library versions, processor type, number of processors used in the run, the use of fully populated nodes or not, processor decomposition, OMP parallelization, hyperthreading, and different configuration and parameters of the input/output (I/O) servers. For this reason, and because of time constraints and several issues with the Met Office CrayXC40 supercomputer during the course of this work (see Appendix), a number of assumptions and simplifications are made.

It was most efficient to study technical performance on the CrayXC40 *Broadwell* nodes only (36 processors per node), as they became the default nodes for research use within the Met Office from 1 September 2016. The alternative *Haswell* nodes (32 processors per node) were used for running initial UKC2 evaluation case studies but are reserved for operational and climate runs in future. Fully populated nodes are used by default with the exception of the I/O servers, which are more memory demanding. The configuration used for this study is as described in Lewis et al. (2017), namely based on the AMM15 domain grid and the **Unified Model 10.1** (UM), **NEMO 3.6 stable** (revision 5518), and **WaveWatchIII 4.18** (WWIII) codes. The default Cray compiler version 5.2.82 and the default compiling options already present in the suites are used, although it might be possible to improve the performance by using the intel or gnu compilers and other compiling options.



© Crown copyright 2014 Met Office

As the UK Environmental Prediction systems at UKC2 are in a Prototype state the configurations and source codes will continue to be updated and developed during Phase 2 activities, only rough quantitative estimations are provided together with recommendations that should be further investigated during Phase 2. Any conclusions about run times differing by less than 5% should be interpreted with care, due to the inherent variability of the CrayXC40 supercomputer. Queuing times associated with different configurations are not considered in the study, as it can be highly variable and not easily measured. In general, it is expected that, the more supercomputer resources one particular run demands, the longer it will stay in queue. This would impact the speed at which research results can be generated for a given test case study, and therefore some consideration on the most appropriate research configuration balancing queue and run times may also be necessary within Phase 2.

There are already some optimization studies for some of the components used in UKC2, in particular for NEMO and the UM but run times are highly dependent on the specific code version and scientific configuration used. When possible, evidence from these previous studies is used to support conclusions drawn from the UKC2 tests described here. Details on the suite and revision numbers used to run each test are provided, so that they will be reproducible. When possible, all runs have been repeated at least three times, and the uncertainty in the run time measures provided estimated as the standard deviation.

## 2. Uncoupled mode

For runs conducted in uncoupled component-only mode the Wallclock time that appears at the Cray End of Job Report is measured, noting that this does not take into account the time needed to run any associated suite scripts but only the model run time. This time takes into account the actual CPU time, the time spent in I/O operation, and time spent in communication. The I/O time is the cause of the greatest run time variability, but it is considered important to be included in the study, particularly when later comparing with coupled runs.

### *2.1 Waves – WaveWatch III vn4.18*

To study the wave component optimization, the UKW2 suite mi-am016 at revision 65964 is run (wind forcing only, cold start).



© Crown copyright 2014 Met Office

It is vital to know the *scalability* of each model, as this will at the end determine the most efficient number of nodes to be used in a run, and the final run time. The scalability gives a measure of how run time changes with the number of processors dedicated to the run. The slowest module of the total coupled system will constrain how fast it can run.

The wave module is the easiest to study in uncoupled mode compared to the ocean and atmosphere modules by different reasons: there is no processor decomposition between the x and the y direction, OMP and MPI parallelization can not be used simultaneously, and there is no input/output server. The only free parameters studied here are the total number of nodes, the use of OMP parallelization, and the use of hyperthreading. *Hyperthreading* is an implementation used to improve parallelization on x86 microprocessors, where for each processor core that is physically present the operational system addresses two virtual cores and shares the workload between them when possible.

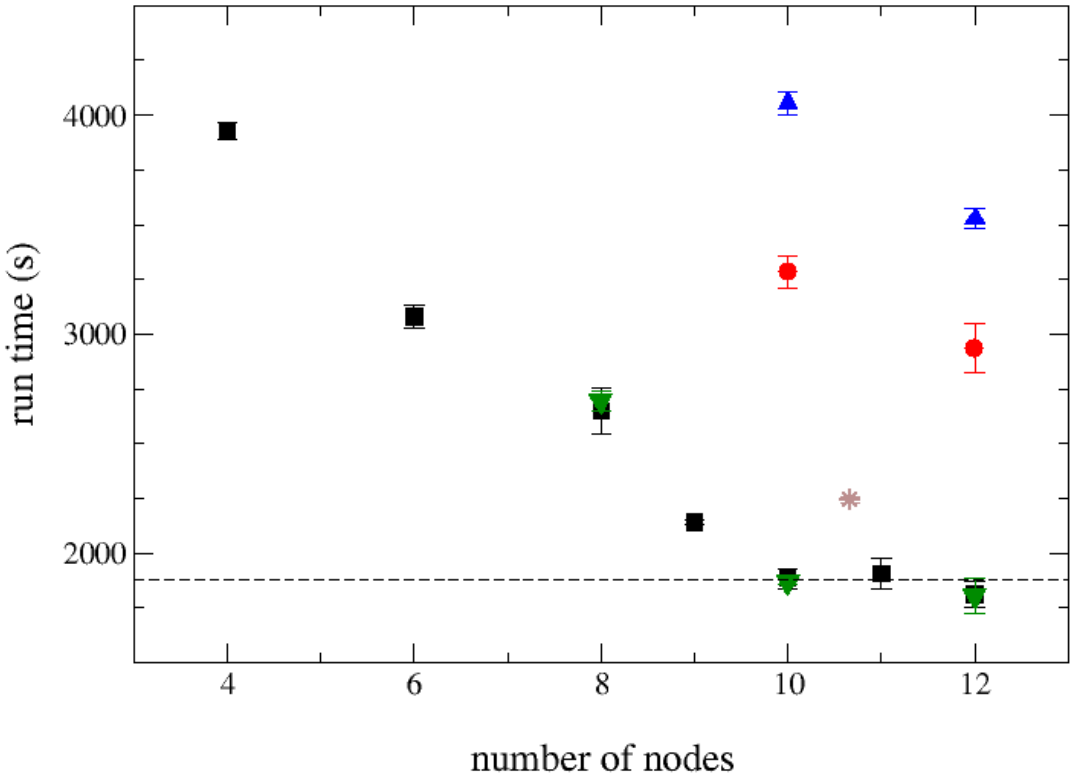


***Figure 1: Run time as a function of physical Broadwell processor nodes for a wave only run (suite mi-am016). Squares, no OMP parallelization, no hyperthreading; circles, 2 OMP threads per node, no hyperthreading; triangles up, 3 OMP threads per node, no hyperthreading; triangles down, no OMP parallelization, hyperthreading; star, original run parameters using Haswell nodes. The dashed line indicates the shorter run time possible using an efficient amount of supercomputer resources.***



© Crown copyright 2014 Met Office

In Figure 1 the model run time is shown as a function of computer nodes used under different conditions. Without OMP parallelization nor hyperthreading (squares), the run time scales linearly with the number of processors until 10 nodes are used, and if more nodes are used the run time does not decrease. In the present configuration, WWIII gives a warning when using 9 nodes or more:

```
*** WAVEWATCH III WARNING IN W3INIT :
    INCREASING TARGET IN MPP PROPAGATION MAP.
    IMBALANCE BETWEEN OVERALL AND CFL TIME STEPS
```

This simply indicates that too many processors are used compared to the spectral decomposition of the system, and therefore the calculation is imbalanced and inefficient, although still valid. The graph shows that the warning is too conservative and that we can still use a few more processors before the run becomes inefficient.

WWIII vn4.18 cannot be run using MPI and OMP parallelization simultaneously. To use MPI, the program must be compiled using the switches MPI and DIST, while to compile with OMP, with the switches OMP0, OMP1, and SHRD. In WWIII vn4.18, OMP parallelization is not completely tested, and when using the switch OMP0 the program crashes when reading one of the input files. On the Cray supercomputer, OMP is used to parallelise work within shared memory nodes. Therefore, the use of pure OMP parallelization in WWIII is not beneficial. As a test, an attempt was made to run the wave suite using MPI and 2 (red circles) and 3 (triangles up) OMP threads per node compiling with the switches MPI and SHRD. The result is that the run time is similar, and even slower, than the run time considering the MPI threads only, showing clearly that OMP parallelization should not be used when running the wave module.

Figure 1 also shows that the performance of runs using hyperthreading (triangles down) is the same as pure MPI runs with the same number of physical nodes. For sake of simplicity, it is therefore recommended not to use hyperthreading for wave runs. Finally, attempts were also made to use hyperthreading and OMP simultaneously without success: if using a low number of nodes, the error message: *“OOM killer terminated this process”* is received, which means that there insufficient memory for the run. If running with 10 nodes or more, the error received is: *'SOMETHING WRONG WITH MPP PROPAGATION MAP'*, meaning that the processor decomposition code is not working properly.



© Crown copyright 2014 Met Office

The original UKW2 wave suite was implemented using 12 Haswell nodes (384 processors) with MPI only, compared to the optimal value shown in Figure 1 of 10 Broadwell nodes (360 processors). This would suggest that, although the original configuration was nearly optimal, at least one node more than the optimal value was likely being used. On examining output from the initial implementation runs, it is found that with the original configuration WWIII was not giving any imbalance warnings, and that the run time was longer than the optimal value for Broadwell nodes (Figure 1, star; the number of equivalent Broadwell nodes was calculated by dividing the total number of Haswell processor by the number of processors in one Broadwell node). This might mean that, although the Broadwell processors are slower than the Haswell, they are more efficient. Possible reasons are the improved memory speed of the Broadwell processors, and having 2 additional cores per socket, which may reduce MPI communication.

In conclusion, the following recommended parameters are found for wave runs:

- **Number of nodes: 10**
- **Pure MPI, no OMP parallelization (switches MPI SHRD)**
- **No hyperthreading**

### *2.2 Ocean – NEMO vn3.6 r5518*

The UKO2 suite mi-al197 is run at revision 65963. First, to study *processor scalability* the same number of processors are used for the East-West (x) and the North-South (y) directions. Figure 2 shows the run time in seconds as a function of the number of nodes (squares). The scalability becomes worse as the number of nodes is increased, but it is still considered reasonable for up to 36 nodes. This behaviour and run time figures are consistent with previous studies using Haswell nodes. If attempting to use a larger number of nodes, a xios (the NEMO I/O server) error occurs, as with xios in attached mode it is not possible to use more processors in a run than the number of meridional grid points, which for the UKO2 configuration is 1345. More nodes could be used if running with xios in detached mode.

In the NEMO version used in this study, OMP parallelization was not implemented (although in the previous runs described NEMO was compiled with OMP options and run with one OMP thread). When using two OMP threads per node (Figure 2, circles), the run time increases by around 30%. If more OMP threads are used, the run time increases even more (Figure 2, triangles up). The same occurs when using



© Crown copyright 2014 Met Office

hyperthreading without OMP (Figure 2, diamonds), where the decrease in run time is similar to the case when using 2 OMP threads. On the other hand, if running with hyperthreading and two OMP threads per node (Figure 2, triangles down), the run times are comparable to pure MPI runs without hyperthreading. For simplicity the study of ocean hyperthreading and OMP is limited to runs having a total of 16 physical nodes, but similar results are expected when using any number of nodes.

The possibility of reducing the run time by changing the *ratio between the processors* in the x and y direction (nx, ny) is explored, while keeping the total number of nodes in the run constant. *Figure 3* shows the run time as a function of this ratio (nx/ny) for a total of 25 (squares) and 36 (circles) nodes (no hyperthreading, no OMP parallelization). In both cases there is a maximum run time when the ratio is one, and a minimum when the ratio is approximately 0.5. With a lower number of total nodes, there is another minimum of similar value when the ratio is 1.5. Therefore, a reduction in run time of around 7% can be expected when the processor decomposition is selected appropriately. These results have to be considered with care though, because the supposed run time reduction is of the same order than the error bars. For nx/ny ratios larger and smaller than those shown in the graph, the runs did not finish within the walltime and crashed with a xios error.

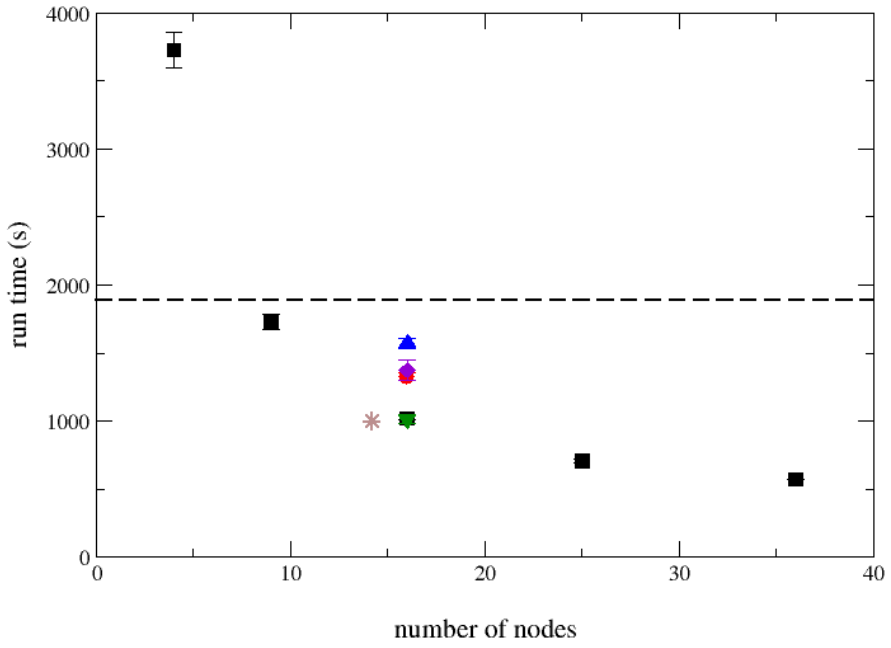


***Figure 2: Run time as a function of physical Broadwell processor nodes for an ocean only run (suite mi-al197) with xios in attached mode. Squares, no OMP parallelization, no hyperthreading; circles, 2 OMP threads per node, no hyperthreading; triangles up, 3 OMP threads per node, no hyperthreading; diamonds, hyperthreading, no OMP parallelization; triangles down, hyperthreading, two OMP threads per node; star, original run parameters using Haswell nodes. The dashed line indicates the shorter wave run time possible using an efficient amount of supercomputer resources. The number of processors in the x and y direction is the same for Broadwell nodes.***



© Crown copyright 2014 Met Office

Finally, the *I/O server* in NEMO is studied, which can be used in attached mode as a library, or in detached mode as an independent program running in its own dedicated processors. As I/O is memory intensive, it is advisable to use non-fully populated nodes, and space the processors in the node. *Figure 4* shows the run time as a function of different xios parameters. Experiment A runs xios in attached mode, while B, C, and D run in detached mode with 12, 18, and 24 dedicated xios processors in the same node. On the basis of these runs it is concluded that xios in attached mode is the most computationally efficient configuration. If using detached mode, similar performance to the attached mode is only obtained when using 18 xios processors at the expense of one extra node, while using less processors slows down the run (13% slower when using 12 xios processors). If more xios processors are used the run slows down again, probably due to less memory available per processor.

Some runs were attempted using centrally maintained and compiled *xios libraries* available on the CrayXC40 (modules cray-netcdf-hdf5parallel/4.3.2, cray-hdf5-parallel/1.8.13, oasis3-mct/1.1/10158/1356, XIOS/1.1/10158/684, XIOS/1.1/10158/684), instead of those compiled as part of local builds for UK Environmental Prediction work. Both in attached (E) and in detached mode using 18 xios processors (F) the runs were slower than using the local xios compilation in attached mode. In fact, the run in detached mode was slower than in attached mode. This could suggest that the local compilation is better optimized, or that the right module combination was not used in testing, but it is difficult to reach definitive conclusions due to the small differences and the size of the error bars.



© Crown copyright 2014 Met Office

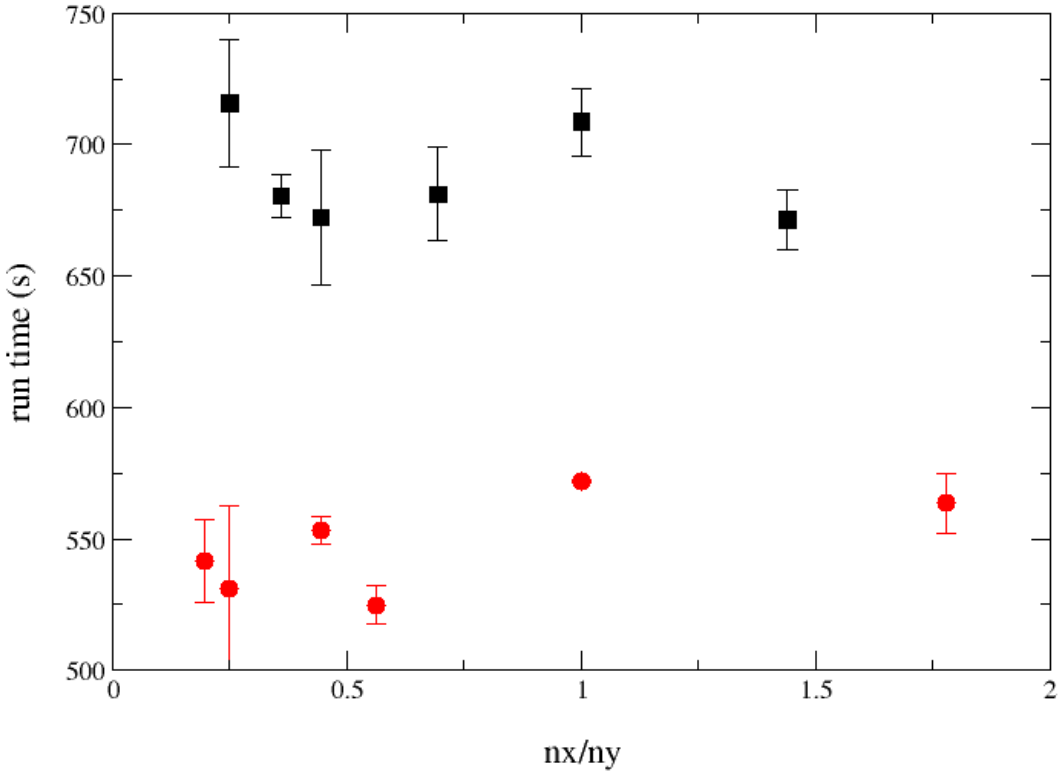


***Figure 3. Run time as a function of the ratio between the processors in the x and y direction (nx/ny) with the same number of physical Broadwell processor nodes in an ocean only run (suite mi-al197) with xios in attached mode. Squares, 25 nodes; circles, 36 nodes. No hyperthreading nor OMP parallelization was used in the runs.***

The run times between runs in the Broadwell nodes and runs with the initial implementation parameters using the Haswell nodes (*Figure 2*, star) are very similar. The Haswell run is in fact faster, but it is using optimal processor decomposition and the Broadwell runs shown in Figure 2 were not. For reference the dashed line in the Figure 2 indicates the minimum wave model run time using an efficient amount of supercomputer resources, showing that the ocean model runs slightly faster with the same number of nodes, and that it can run much faster if more nodes are employed.

In conclusion, the recommended parameters for ocean runs are:

- **Double number of processors in the y direction than in the x direction**
- **Pure MPI, no OMP parallelization**
- **No hyperthreading**
- **Xios in attached mode, multiple output files**

Although the ocean model can run even faster than has been shown here by using more nodes and xios in detached mode, this is not currently considered necessary because the wave model will be the limiting factor for run time when considering coupled models.



© Crown copyright 2014 Met Office

Nonetheless, when running with xios in detached mode it is recommended to use 18 dedicated processors in one node, as this approach maximizes the number of processors and memory available to xios. It is possible to configure xios in order to generate one single output file, instead of one file per processor. Preliminary studies showed that this is highly inefficient, and it should be avoided.

Our results indicate a possible decrease in performance when using the cray modules for the xios libraries, but even though we recommend using them once NEMO is updated to the final 3.6 version (which requires xios 2.0 instead of the 1.0 version we are currently using) during Phase 2 of the project, as they will be operationally supported and they will be the standard at the Met Office.

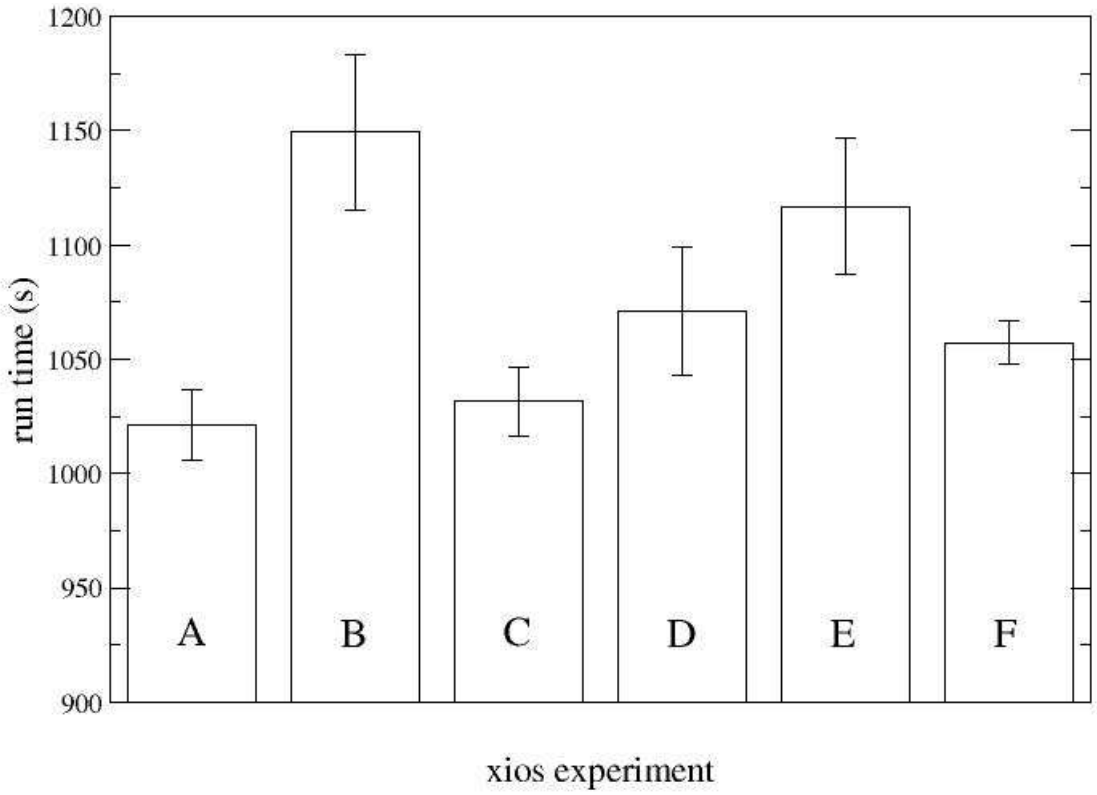


***Figure 4: Run time as a function of NEMO xios configuration in an ocean only run (suite mi-al197), using 16 physical Broadwell processor nodes (plus one extra node for xios in detached mode), the same number of processors in the x and y direction, and no hyperthreading nor OMP parallelization. A, attached mode; B, detached mode with 12 dedicated processors; C, detached mode with 18 dedicated processors; D, detached mode with 24 dedicated processors; E, xios cray module in attached mode; F, xios cray module in detached mode with 18 dedicated processors.***



© Crown copyright 2014 Met Office

### *2.3 Atmosphere – Unified Model vn10.1*

The UKA2 suite mi-al191 is run at revision 66925. The input of Paul Selwood (Met Office) is gratefully acknowledged for providing optimization support for UM. The first tuning can be done with the *segment size*, which is controlled with namelist variables. The initial implementation of the UKA2 suite was based on an operational suite for which such tuning was already done. The following recommendations were therefore implemented:

- use of the 32 bits solver, as it is twice as fast as the 64 bits solver (the 64 bits solver is activated by setting the cpp/fpp compilation flag *C_DP_HML*).
- only writing one pe0 output file, which is controlled with the namelist variable *prnt_writers* (by default we write two output files).
- for version 10.1, if using OMP only 2 threads should be used, as the code is not very well optimized for OMP and it will be slower running with OMP than with pure MPI. On the other hand, it is necessary to run with OMP if running the I/O server. Due to the amount of output generated for UK Environmental Prediction research from the runs, it is possible that the time lost with OMP could be gained using an I/O server. The use of an I/O server in climate application runs is not beneficial, but given that the UKA2 configurations are writing more often to the output this is also tested.
- hyperthreading should only be used with two OMP threads.

There are also some limitations in the UKA2 system: the maximum number of processors permitted in the x direction is 54, and 60 in the y direction, which is a consequence of the grid size; the number of processors in the x direction can not be odd either.

First the UKA2 scalability is studied without the I/O server, and using the same number of processors in the x and y direction. *Figure 5* demonstrates that, in agreement with other experience at the Met Office, the UM model shows a good scalability with a large number of nodes (squares). It can also be seen that the use of two OMP threads per node (circles) greatly decreases the performance, and that the use of hyperthreading and two OMP threads (triangles down) is also not beneficial.

The run time increases as the ratio between the processors in the x and y direction (nx/ny) increases (*Figure 6*). Both for a high and low number of nodes there is a minimum at a ratio of around 0.5, although the results suggest that it might be possible to obtain better performance at an even lower ratio when running with a large number of nodes.



© Crown copyright 2014 Met Office

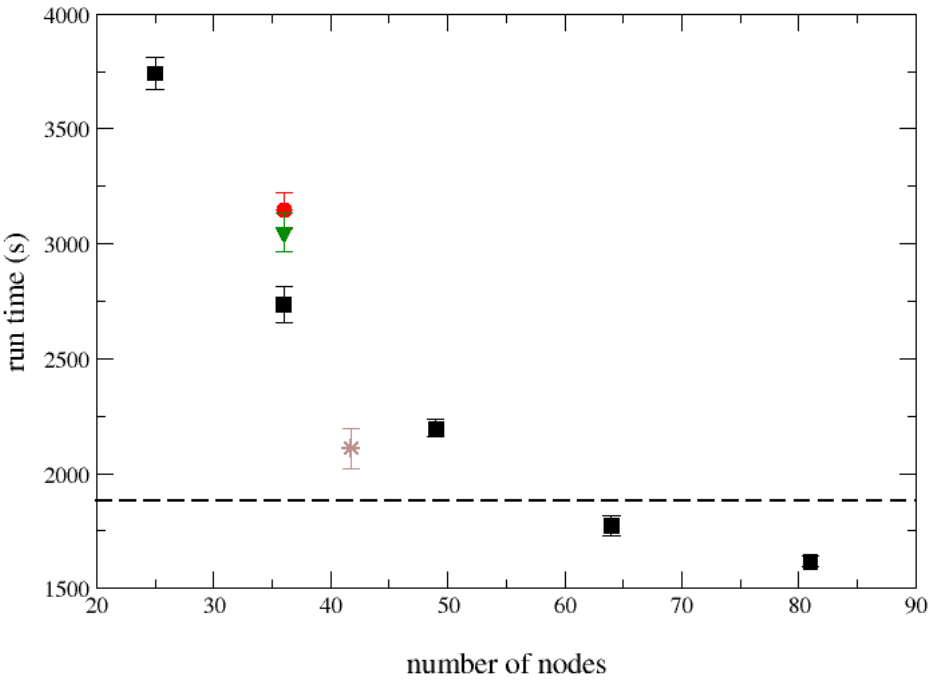


***Figure 5: Run time as a function of physical Broadwell processor nodes used for an atmosphere only run (suite mi-al191) without an I/O processor. Squares, no OMP parallelization, no hyperthreading; circles, 2 OMP threads per node, no hyperthreading; triangles down, hyperthreading, two OMP threads per node; star, original run parameters using Haswell nodes. The dashed line indicates the shorter wave run time possible using an efficient amount of supercomputer resources. The number of processors in the x and y direction is the same for Broadwell nodes.***

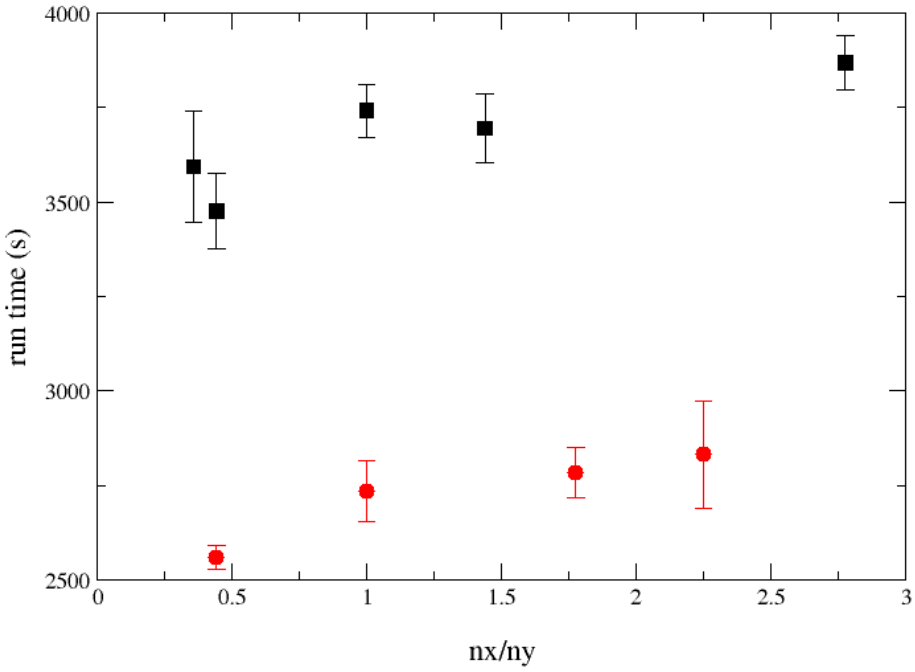


***Figure 6: Run time as a function of the ratio between the processors used in the x and y direction (nx/ny) with the same number of physical Broadwell processor nodes in an atmosphere only run (suite mi-al191) without an I/O processor. Squares, 25 nodes; circles, 36 nodes. No hyperthreading nor OMP parallelization was used in the runs.***



© Crown copyright 2014 Met Office

A number of different tuning options can be performed with the *I/O server*, as described in the UM Documentation Papers C10 manual. The following suggestions are provided:

- enable asynchronous STASH and dump options;
- ensure that the *pe* output is retained in the working directory;
- set diagnostic output to at least operational level;
- remove any existent MPI environment variables and output buffering.
- if any of the servers reached the maximum queue size (can be checked in the output):
- increase IOS buffer size parameter, increase the IOS parallelism, increase the number of IO servers.
- if in the *pe* output there appears the string *"Stall getting asynchronous stash lot at"*,
- increase the number of field aggregated or the concurrency of the asynchronous stash queue.
- adjust the I/O processor decomposition (variable *IOS_spacing* in namelist *IOSCNTL*), so that the I/O server has more memory.
- the total stall time, the time wasted in I/O communication (which is specified at the pe0 file), should be less that 10 seconds (the last I/O communication might be larger due to the closing of output files).
- the variable *IOS_task_per_server* in the namelist *IOSCNTL* selects how many MPI tasks will be used for each I/O server. The total number of tasks used by the I/O server (environment variable *FLUME_IOS_NPROC*) equals that variable times the number of servers. The UM C10 manual explains that having more than 10 I/O servers is usually not advantageous.

*Figure 7* present run time results for different tests for the I/O server, which take into account the possible optimizations that were not present by default in the initial UKA2 configuration. Results have to be considered with caution due to the large error bars and the small differences between run times. Runs using I/O servers with the initial configuration (test B, two I/O servers with eight processors per server), are around 8% faster than without I/O server (test A), although at the expense of one extra node used by the I/O servers. For runs with eight processors per I/O server (test B = 2 I/O servers; test C, 1 I/O server; test D = 3 I/O servers), the best performance seems to occur when using only one I/O server. Writing only one pe output file (test E) seems to slightly reduce the run time compared with the case when we write two output files (test B).



© Crown copyright 2014 Met Office

All the *MPI environment variables* are removed for test F, (MPICH_GNI_MAX_EAGER_MSG_SIZE, MPICH_GNI_MAX_VSHORT_MSG_SIZE, and MPICH_NEMESIS_ASYNC_PROGRESS; note that by removing the variable MPICH_MAX_THREAD_SAFETY the atmosphere module crashes), the run time increases respect the reference case (test B). Test G considers the use of IOBUF, an I/O buffering library that can be linked to the UM code. This library is supposed to reduce the I/O wait time for programs that read or write large files sequentially, by intercepting I/O system calls and adding a layer of buffering, improving program performance by enabling asynchronous prefetching and caching of file data. With this library the run time is slightly reduced, but on the other hand, the library manual states that its behaviour is undefined when used for parallel I/O into a single file, so it should be used with care. There are a series of parameters that can be adjusted when running the IOBUF library, but only the default values were used.

The UM Documentation Papers C10 manual on the I/O server for UM specifies that it is optimal to set the spacing (variable *ios_spacing* of the namelist *IOSCNTL*) such that there is no more than one server per node. For Haswell nodes the optimal value was 32, while for Broadwell nodes it should be 36. This change has been implemented in tests H and I, where the number processors per server and the number of servers are also modified, but keeping the total number of I/O processors equal to 18: test H has two I/O servers with nine processors per server, and test I six I/O servers with three processors per server. The results are comparable to test B and C respectively, resulting in this case in better efficiency when using more I/O servers.

The conclusion regarding the UM I/O server is that, although it is advantageous to use it at the expense of one more node, tuning its different parameters could be a lengthy process that would only provide a small gain. It is therefore recommended to use an I/O server with a basic parameter adjustment.

Run times using the Broadwell nodes and runs with the initial implementation parameters on the Haswell nodes (*Figure 5*, star), are very similar. The Haswell run is indeed faster, but it is using optimal processor decomposition and probably optimal I/O server parameters. The dashed line in Figure 5 indicates the minimum wave model run time found using an efficient amount of supercomputer resources, showing that the atmosphere model runs much slower with the same number of nodes. On the other



© Crown copyright 2014 Met Office

hand, the atmosphere model scales very well with the number of nodes, and can easily run faster than the wave model by dedicating more nodes to it.

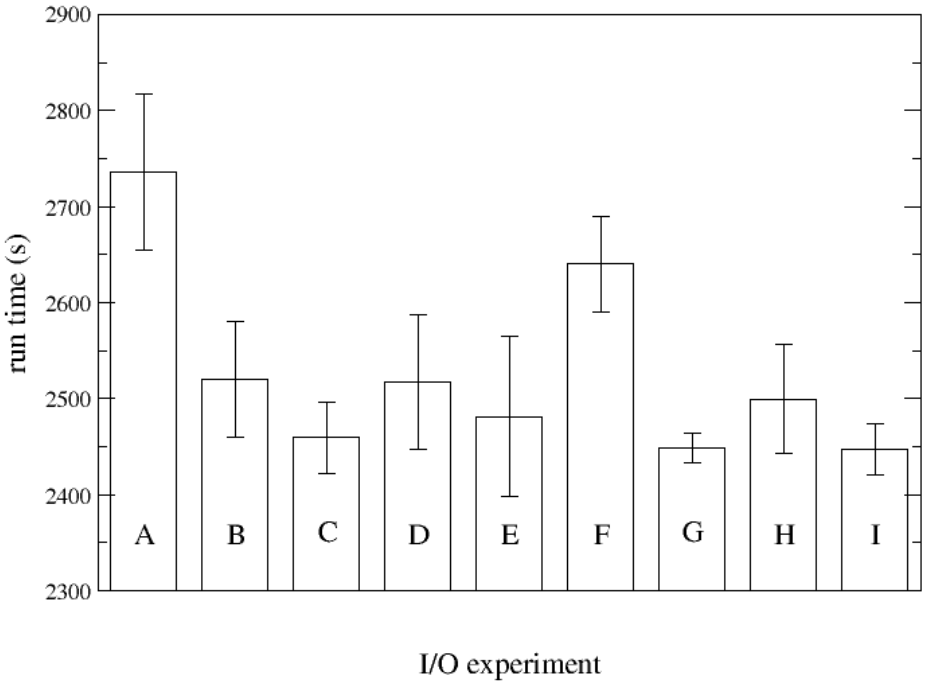


***Figure 7: Run time as a function of UM I/O configuration in an atmosphere only run (suite mi-al191), using 36 physical Broadwell processor nodes (plus one extra node for I/O servers if active), the same number of processors in the x and y direction, and no hyperthreading (and 2 OMP threads per node when I/O active). A, no I/O server; B, 2 I/O servers with 8 processors per server; C, 1 I/O server with 8 processors; D, 3 I/O servers with 8 processors per server; E, 2 I/O servers with 8 processors per server, one pe0 output file; F, 2 I/O servers with 8 processors per server, no MPI instructions; G 2 I/O servers with 8 processors per server and I/O buffering library; H, 2 I/O servers with 9 processors per server; I, 6 I/O servers with 3 processors per server.***

As a conclusion, the recommended parameters for atmosphere runs are:

- **At least double number of processors in the y direction than in the x direction**
- **No hyperthreading**
- **Two OMP threads per node**
- **Use the UM I/O server (ios_spacing=36; ios_tasks_per_server=3; FLUME_IOS_NPROC=18)**
- **One pe output file (prnt_writers=2)**
- **Link with the IOBUF library, default parameter values (IOBUF_PARAMS='*:verbose')**



© Crown copyright 2014 Met Office

## 3. Coupled mode

The main optimization in a coupled run consists in finding, for each module, the optimal running parameters that also makes them run at the same speed. Therefore, the time that one module would have to wait for others to send or receive coupling fields will be reduced. Having studied the behaviour of the different modules that have to be coupled, this optimization for coupled mode should be straightforward and only minor adjustments required. As there is a limit in the number of processors that can be used to increase performance in one given module, the maximum number of processors that can optimally be used is limited by the component module that scales the worst. For coupled mode runs the possibility of running with under populated nodes is now considered, as this would allow a finer tuning of the run time for every module, and it has been suggested that this could speed up the runs (courtesy Met Office CRUM team).

There are two possible approaches to defining the optimal coupled configuration: given a limit in the total number of nodes that can be used in the run, find the optimal processor decomposition for each module; or, considering that there are no constraints in the number of nodes used, find the optimal parameters to run the system in a given time. In this study the second approach is followed, as it is more commonly used when designing operational suites. In particular, an attempt is made to find the parameters that make the suites run faster with the minimum supercomputer resources.

For coupled UKC2 runs the OASIS library is used without debug output, but activating the time statistics. This can be controlled in the namcouple file of the run (EXPORTED instead of EXPOUT; NLOGPRT=0; TIMER_Debug=1, see the OASIS manual for more details). While the atmosphere and the ocean modules have their own timer routines (activated with the namelist variables *l_oasis_timers* in namelist *coupling_control*, and *nn_timing* in namelist *namctl*, respectively), the waves module does not have any, and it is therefore necessary to rely on the OASIS output. The information given by the module timers and the oasis timers is very similar, and it should not matter what type of timer is used for the analysis. The script *manage_times.py*, developed by the Met Office CRUM team, could also be useful to extract the relevant timing information from the atmosphere and ocean output, and work is planned to add an app to the UK Environmental Prediction suites to call it after the run to generate more routine timing information from research runs. Particular focus in this study is paid to the oasis timers *total* (total time of the simulation between the end of routine *oasis_init_comp* and the end of the routine



© Crown copyright 2014 Met Office

*oasis_finalize*), *init_thru_enddef* (reading initial coupling restart and communication of data), *cpl_setup* (time spent in routine *cpl_coupler_setup*), *pmap* (time spent in matrix multiplication in the routine *oasis_advance_run*), *grcv* (time spent in the reception of a field, including waiting time), and *psnd* (time spent in sending a field). The *pmap* time is included in the *grcv* and *psnd* times. There is also some extra time needed to copy a received field to a model variable, but this time is approximately constant and of the order of 0.25 seconds, so is it considered to be negligible here.

### *3.1 Atmosphere – waves coupling*

There are currently no plans to run an atmosphere-wave only coupled suite (i.e. in the absence of an ocean component) for UK Environmental Prediction research in the future. The following discussion is therefore focused on the possible code changes to WWIII that could improve coupling performance. The wave module is the relatively less tested for coupling and it is expected that large optimization measures can be taken in the wave source code.

The high resolution UKC2 atmosphere-ocean-wave coupled suite mi-am057 at revision 66863 is run, using the case study date 20141207. The only coupled fields in the test runs would be the exchange of Charnock coefficient from the wave to atmosphere models and the two wind components from atmosphere to wave. A large number of processors are used for the atmosphere and the ocean modules, to avoid them having to wait for the wave model. Considering the study of the module run times in uncoupled mode, 2 OMP threads, 24x54 UM processors + 16 I/O processors, 15x24 NEMO processors, and 288 WWIII processors are used. In this configuration, the difference in run time for different wave codes will only be related to the changes in the wave code. Furthermore, the coupling will be faster the less time the atmosphere module waits to receive information from the wave module.

A number of different changes have been implemented to the wave code concerning coupling, which are analysed independently in the following discussion.

#### General improvements

These improvements consist of the elimination of some extra variables that had to be allocated during coupling, several bug fixes, and the MPI distribution of the received coupled fields only if they have been successfully received, instead of doing it in every time step. The differences between the original code and the improved version is



© Crown copyright 2014 Met Office

documented in the differences between revisions 988 and 1328 of the wave branch svn://fcm3/WW3_svn/WW3v4/branches/dev/frhl/r966_ww3v4_ukep/model. Note that the format of the WWIII restart and template files change between these revisions, and they would need be adapted accordingly during testing

### Vector vs. Matrix send/receive

In the original WWIII coupling branch (r988), the fields sent and received via OASIS were defined as matrixes, with dimensions (NSEA, 1) for NSEA valid sea points in the wave grid. The WWIII variables are stored in vectors containing only the information of the NSEA ocean points in the grid. Therefore, when one variable is coupled, first it has to be copied to a matrix, and then the matrix is sent to the coupler. It is instead much more efficient to declare the OASIS variables as vectors, and send and receive the WWIII variables via OASIS directly. In this way, the allocation of the intermediate matrixes and copying the fields to the matrix variables can be avoided. This change is provided in revision 1369 of the WWIII coupling branch, which also includes the general improvements of revision 1328.

### Points partition

Originally, all the communication with OASIS is achieved via one single processor in the wave module (a *serial partition*). The disadvantage of this approach is that, in every time step, one single processor gathers all the data that has to be transferred from all processors, then only this processor data calls OASIS, which will decide if the field needs to be actually transferred in the current time step or not. When receiving a field, one single processor will call OASIS, which will receive the field at the appropriate time step, and then this processor will communicate the received field to the rest of the processors.

It would be optimal if the communication were done directly via OASIS using all processors, avoiding the need to gather and scatter the received and sent fields, but rather every processor will send and receive the data it handles directly thanks to the remap files. This means that the MPI communication is only between coupling modules, while communication between different processors in the wave module is not needed any more for performing the coupling. Furthermore, communication only takes place if any field has to be transferred, instead of at every time step. In this way, the amount of MPI calls is greatly reduced.



© Crown copyright 2014 Met Office

The data in the wave module is stored as a vector of wet points. The processor partition of this vector between different processors is as follows:

```
DO I=1,NAPROC
   NNSEAL=1 + (NSEA-I)/NAPROC
   DO JSEA=1,NNSEAL
    ISEA=I + (JSEA-1)*NAPROC
   END DO
  END IF
END DO
```

In this code, *ISEA* is the index of the vector, which goes from 1 to *NSEA*, the number of wet points. *NAPROC* is the number of processors and *I* the processor number. *JSEA* is the index of the vector stored in one processor, which goes from 1 to *NNSEAL*, the number of points stored in that processor. With this in mind, it is easy to demonstrate that the vector stored in every processor has wet point indexes separated by *NAPROC*. Therefore, in OASIS a point partition has to be defined for each variable that has to be transferred (see the OASIS manual for further details). This code modification is currently in development and will be assessed further as part of ongoing optimisation within Phase 2.

In all tests some of the OASIS timers are similar for both components, in particular *oasis_init_comp* (~60 s) and *cpl_setup* (~10 s). This indicates that the initial setup of the coupler approximately takes the same amount of time independently of the model and the details of the run configuration. It is expected that this coupling setup takes place simultaneously in all models, so that this extra time only has to be taken into account once. The atmosphere and wave *total* times are also very similar, as the models have to wait for each other in order to finish the run. On the other hand, *pmap* takes ~2.3 s for the wave module and just a few milliseconds for the atmosphere module. This difference is probably due to the difference in rank between variables in both models, as the variables in the atmosphere model are matrixes and in the wave module vectors. As expected, the *psnd* times are very small, of the order of milliseconds, as the put routines in OASIS are non-blocking. Surprisingly, although the jobs were set in a way that the atmosphere model run much faster so that the wave model does not have to wait to receive the atmosphere fields, there is occasionally a waiting time in the wave model *grcv* of up to 40 s, although it is usually below 1 s. This probably indicates that during



© Crown copyright 2014 Met Office

some steps of the simulation the atmosphere model may run a bit slower that the wave model, generating a small run time overhead. Therefore, there will always be a certain time spent in one model waiting for another to receive data, no matter how well the processor decomposition is tuned for each model.

In *Figure 8* the atmosphere *total* (left) and *grcv* times (right) are shown as a function of the wave coupling code. Differences in *total* times can be attributed to differences in the initialization and finalization routines that are run before and after the initialization and finalization coupling routines. The wave code with general improvements (centre bar) clearly reduces the run time in around 10% and the atmosphere waiting time in around 25%, indicating that the general improvements greatly increase the performance of wave coupling without much effort. As one of these improvements was to reduce in 90% the calls to one MPI communicator during coupling (with our current coupling frequency), it is expected that the points decomposition will greatly increase the performance of wave coupling. On the other hand, the gains of the vector send/receive code (right bar in each plot) are not so clear, as the run time seems to increase with respect to the 'general improvements' code. The atmosphere receive time slightly decreases, but the error bars are too large to reach definitive conclusions. Nonetheless, the gain to be expected with the vector code, if any, is small.

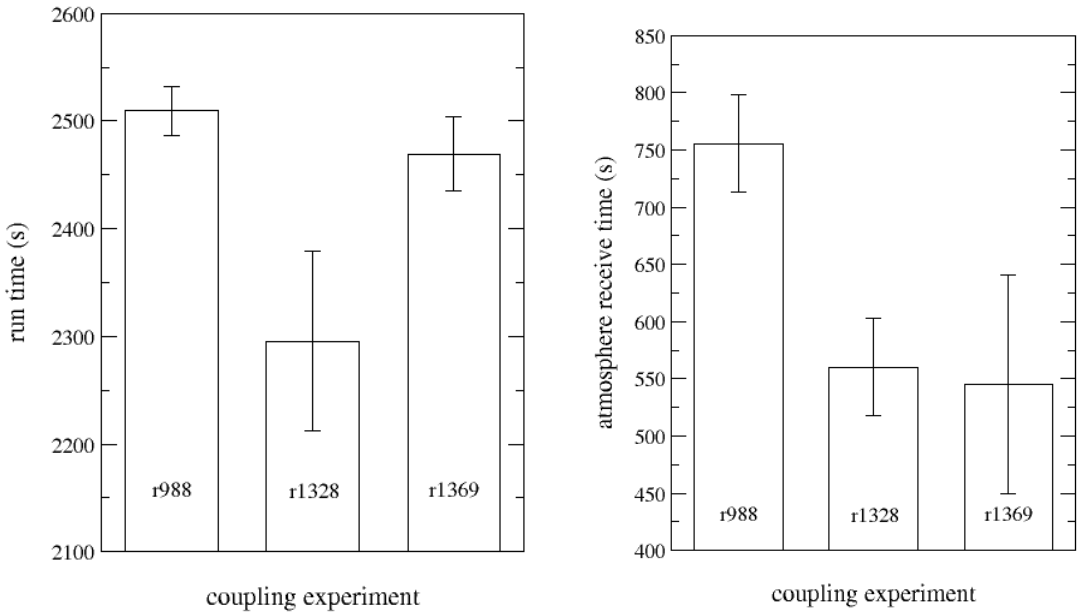


***Figure 8: Atmosphere run time (total time, left) and receive time (grcv time, right) as a function of the wave code revision in an atmosphere/wave coupled run (modified suite mi-am057): left bars, revision 988 (initial wave code); centre bars, revision 1328 (general improvements); right bars, revision 1369 (vector send/receive). See text for differences between revisions.***



© Crown copyright 2014 Met Office

Some initial conclusions can be made regarding atmosphere/wave coupling:

- **There will be always an approximately constant extra time needed for coupling setup**
- **The cost of sending data via the coupler is negligible**
- **The cost of receiving data via the coupler can not be decreased beyond a certain value even at optimal model processor decomposition**
- **The general improvements in the wave code greatly reduce the coupling time**
- **The points partition approach in the wave code should be investigated during Phase 2**

### *3.2 Atmosphere – ocean coupling*

The ocean feedback UKC2ao suite mi-al842 at revision 68990 is run. In this suite, there are 12 coupled fields: three fields sent from the ocean to the atmosphere model, and nine sent from the atmosphere to the ocean model. The number of nodes in both the atmosphere and ocean model is selected such that they run at a similar speed or higher to the maximum speed that is achievable in the wave mode using optimal parameters. It might be possible to run the atmosphere/ocean coupled suite faster, but for later comparison it would be useful to have conditions similar to the ones that will be used in the fully coupled atmosphere-ocean-wave system, in which the wave module is the time limiting factor (e.g. Figure 2, Figure 5).

A first optimization is the use of OMP threads for the atmosphere but not for the ocean module, as it was found that the performance of the ocean module actually decreases when using OMP. Initially this was not possible in the UKC2 coupled suite, as the use of OMP was set to be the same in all modules.

*Figure 9* compares the atmosphere *total* time (left), the atmosphere *grcv* time (centre), and the ocean *grcv* time (right) between two atmosphere/ocean coupled runs with the same number of nodes for the atmosphere model (2 OMP threads, 18x49 MPI processes = 49 nodes) and the ocean model (9 nodes), where the ocean model is run either with 2 OMP threads (left bars, 9x18 MPI processes), or with 1 OMP thread (right bars, 12x27 MPI processes). Although it was found that the ocean model runs faster without OMP parallelization than with 2 OMP threads per node for a given number of nodes, the atmosphere-ocean coupled system runs ~10% faster when using 2 OMP threads per node for the ocean module (*Figure 9*, left). The explanation is that, when



© Crown copyright 2014 Met Office

running without OMP, the ocean module runs faster than the atmosphere model, and it has to wait much longer for it to receive data than when it runs slower with 2 OMP threads per node (*Figure 9*, right): the initial implementation of the coupled system was therefore not well load balanced. When running the ocean module with 2 OMP threads per node the receive times of the atmosphere and the ocean models are of the same order of magnitude, confirming that even with optimal load balance there will always be a minimum waiting time for receiving fields between coupled models. It is noteworthy that once the first field is received in the model, the time needed to receive the rest of the received fields is just a few milliseconds, indicating that the coupling cost does not depend much on the number of fields exchanged.

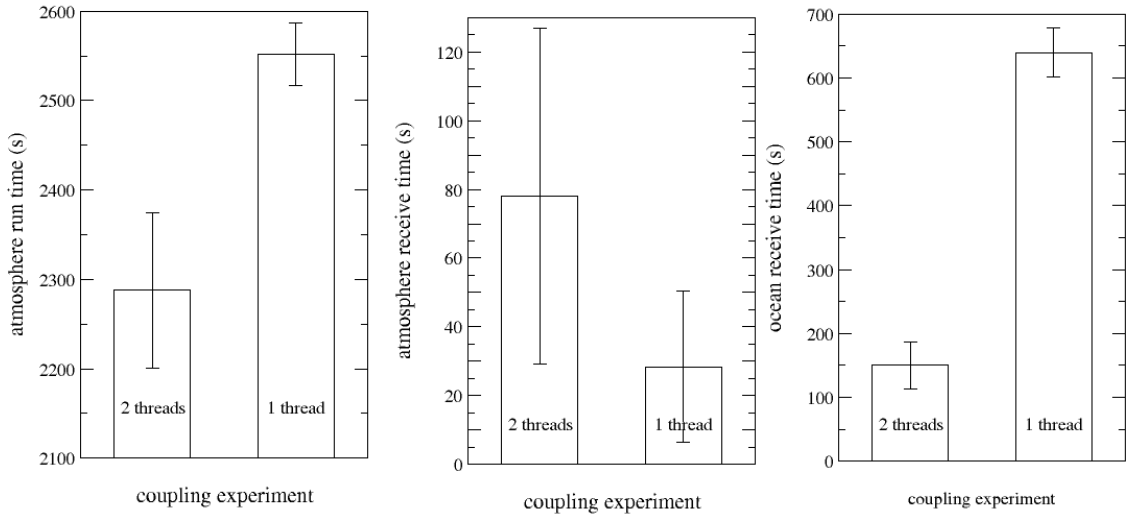


***Figure 9. Atmosphere run time (total time, left), atmosphere receive time (atmosphere grcv time, centre), and ocean receive time (ocean grcv time, right) in an atmosphere/ocean coupled run (suite mi-al842), with ocean running with 2 OMP threads per node (left bars) and with 1 OMP thread (right bars).***

The mi-al842 atmosphere-ocean coupled suite was originally running using a total of 2528 Haswell processors excluding the I/O servers (2*752 for UM, 2*512 for NEMO), compared to 2088 Broadwell processors in the configuration previously used (2*882 for UM, 2*162 for NEMO). Despite using 440 processors less in the Broadwell configuration than in the Haswell configuration, the runs were ~400 s faster. The difference arises from the fact that in the Broadwell configuration the atmosphere and ocean models had a proper load balance, while in the Haswell configuration the ocean model ran much faster than the atmosphere. Therefore, the ocean model wasted time waiting for the atmosphere model to catch up, being the ocean *grcv* time ~1800 s (and the atmosphere *grcv* time no more than 40 seconds). This result highlights the importance of load balance in coupled models.



© Crown copyright 2014 Met Office

A few conclusions can be made concerning the atmosphere-ocean coupled system:

- **Load balance is the main factor to reduce run time in coupled systems given a fixed number of supercomputer resources**
- **It is possible to have a good idea about the right load balance for an atmosphere/ocean coupled system by looking at the uncoupled run times**
- **There is always a minimum waiting time for receiving fields between coupled models even in well balanced systems**
- **It should be possible to fine tune the load balance using unpopulated nodes**
- **The coupling cost does not depend much on the number of fields exchanged**

### *3.3 Ocean – waves coupling*

The partially coupled UKC2ow ocean-wave coupled suite mi-am029 at revision 69782 (wave code revision 1328) is run. This is an interesting system in the sense that fields are only sent from the ocean to the wave model, which can give us an idea of the minimum *grcv* time. In this configuration it would be important to make sure that the ocean model runs faster than the wave model.

In *Figure 10* the difference in run time for the ocean and wave models is compared in both coupled (right bar) and uncoupled model (left bar), dedicating in all cases nine nodes to each model without OMP parallelization. It is noteworthy that the ocean model (*Figure 10*, left) looks to run ~7% faster in coupled mode than in uncoupled mode, but this is because in the coupled mode tests an optimal processor decomposition for the ocean component (double the number of processors dedicated to the y direction than the x direction) was used, while it was not in uncoupled mode. Furthermore, the coupled *total* time does not include the time spent in routines running before coupling initialization and after coupling finalization, while the uncoupled run time includes the total run time. Nevertheless, the ocean run time graph demonstrates that the run time difference between the ocean running coupled or uncoupled codes is minimal, taking into account that as in these coupled runs the ocean model is only calling OASIS put routines and they are non-blocking.

The situation is completely different for the wave model (*Figure 10*, right), as the wave model runs ~30% slower in coupled (right bar) than in uncoupled mode (left bar). This happens even when the *grcv* time is usually below one second, so that the wave model



© Crown copyright 2014 Met Office

does not need to wait for the ocean model to receive data. On the other hand, the absolute value of the run time increase with respect to the uncoupled run is only slightly larger than in the atmosphere-ocean coupled case. This might indicate that the wave coupling code is not very efficient and it should be reviewed and optimized as a priority. Large gains could likely be obtained if internal MPI calls were removed from the wave code during coupling by implementing a 'points partition' coupling scheme. This modification is currently under development and testing.

The original mi-am029 implementation was running using a total of 1024 Haswell processors, compared to the Broadwell test running with a total of 648 processors. Both configurations resulted in similar run times, with the Broadwell configuration running with 376 less processors. It can also be inferred that it is possible to run the coupled suite with Broadwell processors using fewer nodes for the ocean module without any change in the run time, as the ocean runs much faster than the wave module in coupled mode. Seven ocean nodes seems a sensible number after observing the run times for uncoupled ocean runs, making a large overall saving in computational resources (448 processors) compared with the original Haswell configuration.

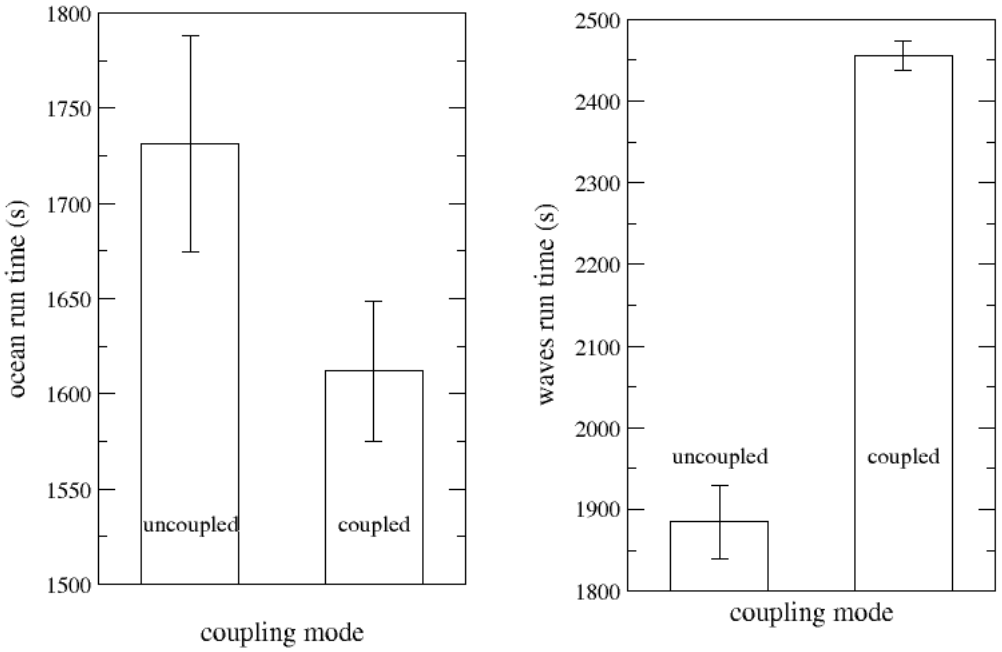


***Figure 10. Ocean (left) and wave (right) run times (total time) in uncoupled runs (left bars, suites mi-al197 and mi-am016 respectively), and an ocean/wave coupled run (right bars, suite mi-am029) using nine nodes for each model in both coupled and uncoupled mode.***

The main conclusions that can be made are:



© Crown copyright 2014 Met Office

- **There is always a minimum waiting time for receiving fields between coupled models**
- **The performance of the wave module in coupled mode is very poor compared to uncoupled mode: code optimization should be considered**
- **The number of ocean dedicated nodes can be reduced more than expected in ocean/wave coupled runs without impact in performance**

### *3.4 Atmosphere - ocean – waves coupling*

Finally the fully coupled UKC2aow suite mi-am057 at revision 70019 (wave code at revision 1328) is considered. There are 18 coupled fields that have to be transferred in order: one field from wave to atmosphere; three from ocean to atmosphere; three from ocean to wave; nine from atmosphere to ocean; and two from atmosphere to wave. The configuration is therefore more complicated than the previous coupled suites, as every model is communicating and waiting for information coming from other two models. Therefore, for each model (except for ocean) there are two significant *grcv* times, one for each model from which it receives data.

The *total* time for the different components of the atmosphere-ocean-wave coupled suite (*Figure 11*, left) using a first estimation of the optimal running parameters for each model based on the scalability of the uncoupled models (atmosphere: 2x18x49 processors; ocean: 12x24 processors; waves: 360 processors) is the same within error bars. This could suggest that the system is properly balanced. On the other hand, according to the *grcv* times (*Figure 11*, right) this would seem not to be the case. The wave model is waiting for a long time to receive fields from the ocean model, and by a smaller amount also for the atmosphere model. The rest of the receive times are below 100 s, in particular for the atmosphere-ocean communications, as would be expected of a reasonably well balanced system. It is concluded that the wave model is running fast compared to the other models, and in particular to the ocean model within UKC2aow. Therefore, it might be convenient to increase the speed of the ocean model so that the wave model does not have to wait for it, which is very cheap because the ocean model scales very well. It might also be convenient to under populate some nodes for fine tuning the load balance, although this would not be very important if the ocean and wave models run faster than the atmosphere as they are relatively very cheap compared to the UM atmosphere component.



© Crown copyright 2014 Met Office

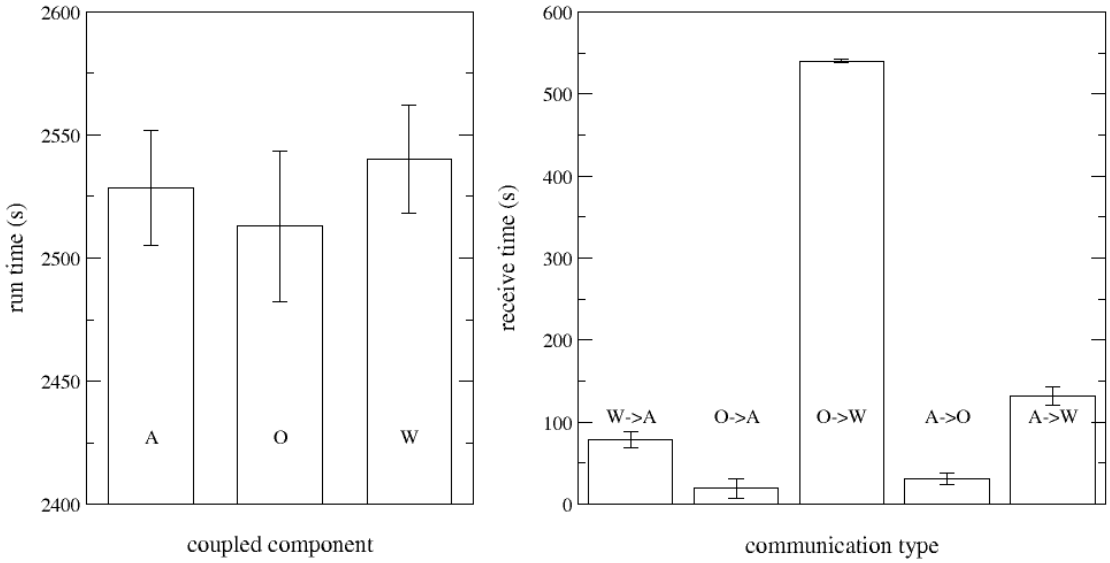


***Figure 11. Run time (total time, leftt) and grcv time (right) in an atmosphere/ocean/wave coupled run (suite mi-am057), with atmosphere running with 2 OMP threads per node and ocean and waves without OMP. A, atmosphere; O, ocean; W, waves. The arrows in the grcv graph indicate the time needed to receive data sent from one particular model to another.***

The original implementation of suite mi-am057 (e.g. as used for initial research case study runs) was using 886 processors more in the Haswell configuration than in the current quasi-optimal Broadwell configuration, and was running ~20% slower. This highlights again the importance of load balance in coupled systems to reduce computational resources and time, and the potential savings that can be delivered given investment in technical optimisation and improvement.

The main conclusion that can be draw after studying the UKC2aow system is that:

- **It may be convenient to make the ocean and wave models run slightly faster than the atmosphere model in an atmosphere-ocean-wave coupled run, as the atmosphere model is the most expensive component**

## 4. The 'cost' of coupling

A key initial technical question in the context of building and evaluating the Prototype high resolution coupled UK Environmental Prediction system was on the technical feasibility in terms of performance and whether the overhead or 'cost' of coupling between components was restrictive for immediate research purposes, and looking further ahead to the feasibility within operational systems.



© Crown copyright 2014 Met Office

There are at least two dimensions that need to be addressed in order to answer the question: *what is the cost of coupling of the UK Environmental Prediction coupled suites?* First, it is possible to consider the supercomputing resources (number of nodes) and the run time of a reasonably optimized suite coupling different components, and provide a comparison of it with the run time of those suites running the same components in uncoupled mode and with the same supercomputer resources. This approach reveals the time cost of running the same models to deliver the same total outputs in coupled or uncoupled mode, given an assumed finite supercomputing resource. A number of conclusions can already be drawn relating to this from the preceding optimization study, which are detailed in the Summary below. The second approach to discussing the cost of coupling is to consider the maximum speed at which a coupled configuration can run, which will depends on the slowest component of the coupled configuration, and compare it with the minimum run time that can be achieved for the different components in uncoupled mode, independently of the supercomputer resources required and so assuming there is no constraint on the total supercomputing resource potentially available.

*Figure 12* contains a summary of the optimization results obtained to date, which enables some conclusions to be drawn regarding optimization and the cost of coupling. First, it is clear to see how even a quick optimization procedure (UK2 runs – representing optimised configurations) can easily reduce the run costs compared to 'out-of-the-box' initial configurations (UK2-H runs – representing initial Haswell configurations). For all uncoupled suites considered UKA2, UKO2 and UKW2a at least a reduction of run time or number of processors used has been achieved through optimisation. For the case of all coupled suites UKC2ao, UKC2ow and UKC2aow (red symbols in Figure 12), reductions in *both* the run time and number of processors has been delivered.

In *Figure 12* the number of nodes used in a coupled run is the same as the sum of nodes used for running its model components in uncoupled mode (both running with the same, nearly-optimal parameters). This means that there is no extra cost of coupling in terms of the total number of nodes when running a coupled simulation and its components separately in uncoupled mode, given an assumption that output from all component systems are required. On the other hand, even with this approach there is a cost in terms of run time with respect to the slowest component of the run, which is



© Crown copyright 2014 Met Office

found to be ~400 s when coupling 2 models, and ~600 s when coupling 3 models. It may be possible to reduce these times with finer optimizations and coupling code changes. This cost should remain nearly constant when using a different number of total nodes and a load balance of the same quality, as it is associated with the initialization OASIS coupler routines and the *grcv* times (the latter mainly depending on the load balance).

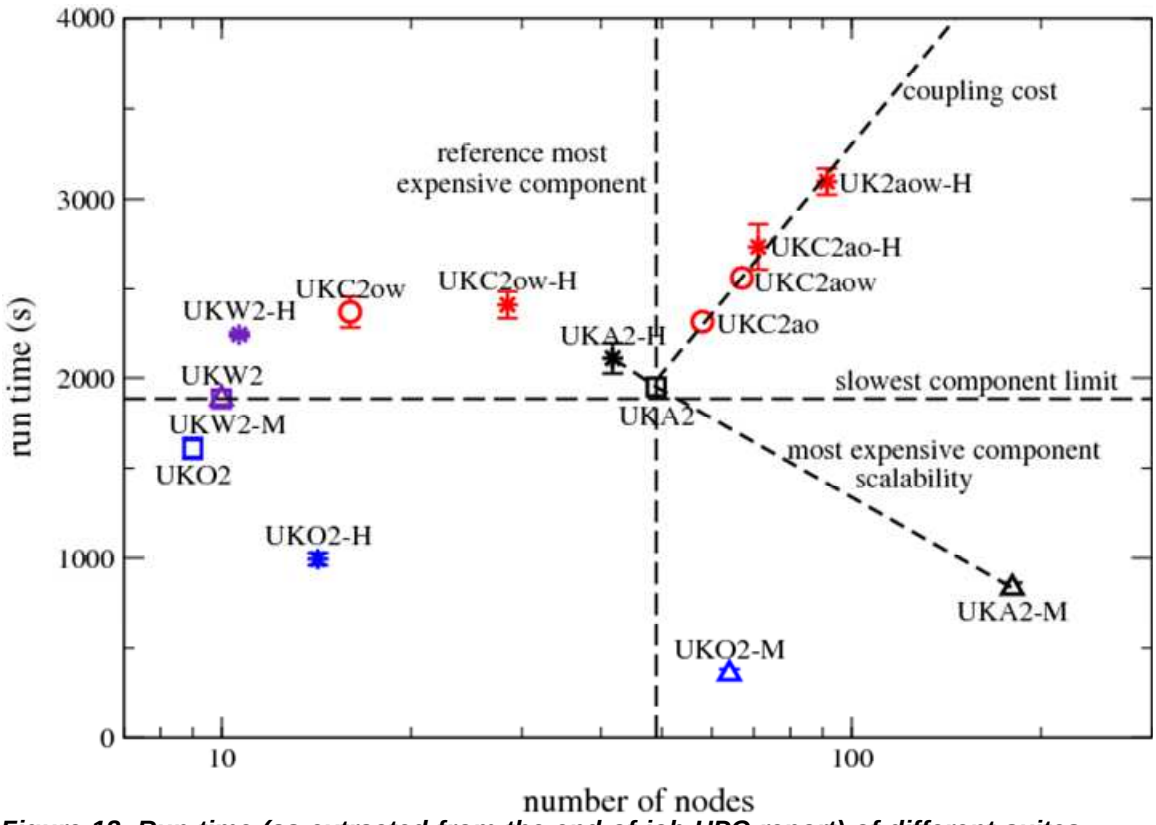


***Figure 12. Run time (as extracted from the end-of-job HPC report) of different suites as a function of the total number of Broadwell nodes. Black, atmosphere only runs; blue, ocean only runs; violet, wave only runs; red, coupled runs. Squares, runs using the nearly-optimal parameters found in this study; triangles, runs using the maximum number of supercomputer resources possible; stars, runs using the original, non-optimized Haswell parameters. The run parameters for the components of a coupled run (number of nodes, OMP parallelization, I/O server configuration, etc) are the same than in the uncoupled run. Each run is described by its UKEP name, with and ending H meaning runs with the original Haswell parameters, and an ending M meaning maximum number of supercomputer resources possible for a model component. Dashed lines in the graph are described in more detail in the text. Note the x-axis is logarithmic.***

Taking into account the most expensive component of the coupled system as a reference, in terms of requiring more nodes to run at a given speed, a trend can be seen in the coupling cost. The run times of all coupled configurations containing the most expensive (i.e. atmosphere) component lie on a line (noting the logarithmic scale on the x axis), even if the configuration is not properly balanced. This line of positive gradient is marked in *Figure 12* with a dashed line and labelled '*coupling cost*'. Therefore, it should



© Crown copyright 2014 Met Office

be possible in principle to estimate the coupling cost of a given run, knowing the most expensive component of the coupled model, and the total nodes used in the run.

Given an unlimited number of nodes available, the maximum speed at which a coupled suite can run is limited by the maximum speed of the slowest coupled component. In the current UKC2 system the slowest component is the wave model, and maximum speed at which it can run is marked in *Figure 12* by the horizontal line labelled '*slowest component limit*'. Therefore, in order to run the UK Environmental Prediction coupled research suites faster it will be necessary for improved scalability of the wave component. For example, there are plans to run the UK high resolution wave model using a Spherical Multiple Cell (SMC) grid with the same 1.5 km resolution as currently used in UKW2 close to the coast, but varying to reduced resolution (e.g. 3 km) further to the open ocean. Coupling and remapping fields between this grid and the atmosphere and ocean grids should be investigated within Phase 2, given the potentially significant gains in system run time that this could deliver, as well as ensuring consistency and efficiency with use of (pre-)operational component modules for research. Were the speed of the wave model to be increased, the horizontal '*slowest component limit*' line in Figure 12 would move down. The limiting slowest component would then very likely be the atmosphere. Given no constraints on number of processors available, it could be envisaged that the slowest component limit line (and therefore the origin of the '*coupling cost*' line) would reduce to the level of UKA2-M at around 800s. In practice, some balance would likely need to be struck between run time and total number of processors according to the '*most expensive component scalability'* curve, and the origin of the 'cost of coupling' line would be shifted accordingly.

Without being constrained by the slowest component limit, were components to be run completely independently of each other for example (as currently for operational regional atmosphere, ocean and wave systems), it would of course be possible to complete component runs much faster than in coupled mode, albeit using more nodes. The minimum achievable run times for each model are marked in *Figure 12* with the label UK2-M. It could be possible to run the atmosphere and ocean uncoupled models in more than half the time than the minimum run time possible of the coupled suite, by using 180 and 64 nodes respectively. On the other hand, for the existing UKW2 wave component (using the regular high resolution grid) the minimum run time achieved with only 10 nodes is only around 25% faster than the fully coupled atmosphere – ocean – wave run.



© Crown copyright 2014 Met Office

## 5. Conclusions and outlook

This Research Note describes a comprehensive study of the technical performance of the UKC2 system and its component models. The following conclusions are drawn.

- The coupling cost in terms of time mainly depends on load balance
- The coupling cost in terms of computer nodes depends on the scalability of the coupled components
- The coupling cost in terms of time should be fairly constant given optimal load balance conditions, and independent of the total number of nodes used in the run
- The coupling cost in terms of time does not increase much with the number of fields exchanged between models, and should be checked further during Phase 2
- The influence of coupling frequency on the coupling cost will also be investigated in Phase 2

- The coupling cost in terms of additional run time for the UK Environmental Prediction UKC2 system is ~400s (order 20%) for a 2-way and ~600s (order 30%) for a 3 way coupled system
- The maximum speed at which one coupled model can run is critically limited by the maximum speed of the slowest component
- In Phase 2 it would be advisable to investigate methods to improve the wave model scalability, for example by attempting to couple to the SMC grid wave model configuration. This is expected to deliver substantial improvements in run times beyond those reported in this work.
- It would be interesting to study in Phase 2 possible improvements to the wave coupling code, for example eliminating internal MPI communication during coupling by implementing the ‘points partition’



© Crown copyright 2014 Met Office

**Met Office**
FitzRoy Road, Exeter
Devon EX1 3PB
United Kingdom

Tel (UK): 0370 900 0100 (Int): +44 1392 885680
Fax (UK): 0370 900 5050 (Int): +44 1392 885681
enquiries@metoffice.gov.uk
www.metoffice.gov.uk